

Ising formulations of many NP problems

Andrew Lucas

Department of Physics, Harvard University

We provide Ising formulations for many NP-complete and NP-hard problems, including all of Karp's 21 NP-complete problems. This collects and extends classic results relating partitioning problems to Ising spin glasses, as well as work describing exact covering algorithms and satisfiability. In each case, the state space is at most polynomial in the size of the problem, as is the number of terms in the Hamiltonian. This work may be useful in designing adiabatic quantum optimization algorithms.

Correspond with: lucas@fas.harvard.edu

February 26, 2013

Contents

1	Introduction	2
2	Partitioning Problems	3
2.1	Number Partitioning	3
2.2	Graph Partitioning	4
2.3	Cliques	4
3	Hard Constraint Covering and Packing Problems	5
3.1	Binary Integer Linear Programming	5
3.2	Exact Cover	5
3.3	Set Packing	6
3.4	Vertex Cover	6
3.5	Satisfiability	7
3.6	Minimax Graph Problem	7
4	Soft Constraint Packing and Covering Problems	8
4.1	Set Cover	8
4.2	Knapsack with Integer Weights	9
5	Coloring Problems	9
5.1	Graph Coloring	9
5.2	Clique Cover	10
5.3	Job Sequencing with Integer Lengths	10
6	Hamiltonian Cycles	11
6.1	Hamiltonian Cycles and Paths	11
6.2	Traveling Salesman	11

7	Tree Problems	12
7.1	Minimal Spanning Tree with a Maximal Degree Constraint	12
7.2	Steiner Trees	13
7.3	Directed Feedback Vertex Set	13
7.4	Undirected Feedback Vertex Set	14
7.5	Feedback Edge Set	15
8	Graph Isomorphisms	15
9	Conclusion	16
	References	16

1. Introduction

Recently, there has been much theoretical interest in the possibility of using an adiabatic quantum optimization technique to solve NP-complete and NP-hard problems [1, 5]. This is due to the following trick: suppose we have a quantum Hamiltonian H_P whose ground state corresponds to the solution of a problem of interest, and another Hamiltonian H_0 , whose ground state is “easy” (both to find and to prepare in an experimental setup). Then, if we prepare a quantum system to be in the ground state of H_0 , and then adiabatically change the Hamiltonian for a time T according to

$$H(t) = \left(1 - \frac{t}{T}\right) H_0 + \frac{t}{T} H_P, \quad (1)$$

then if T is large enough, the quantum system will remain in the ground state for all times, and thus at time T will correspond to a solution of an NP problem.

In this paper, we will focus on the first part of the problem: finding a quantum Hamiltonian, H_P , which can encode the ground state of a problem of interest. In particular, we will look for *classical Ising* Hamiltonians, which are extremely simple to interpret as quantum Hamiltonians by simply converting each classical spin variable into a qubit, and have the added advantage that they may be more adaptable to quantum computing hardware. In each case, there are an infinite number of possible choices of H_P which can be used, although many of them will be simply related to the others by, for example, slightly tweaking some couplings or magnetic fields. We will not ask the question of how robust the Hamiltonians are to slight changes in their couplings or magnetic fields, and simply content ourselves with finding a single Hamiltonian for each problem.

Analogies between Ising Hamiltonians and NP problems have been frequently studied in the past [8, 6], although many of the NP problems previously studied are very straightforward to phrase in this manner. This paper can be thought of as the result of a challenge to a theoretical physicist to show how “all of the famous NP problems”¹ [7] can be written down as Ising models. The “subtle” techniques in this paper are primarily of a few flavors, which roughly correspond to the tackling the following three issues through a (polynomial-size) expansion of the state space: problems with inequalities as constraints (for example, $n \geq 1$, as opposed to $n = 1$), problems where the natural variables have q states (with $q > 2$), and problems which ask global questions about graphs, such as connectivity. The methods we use to tackle these problems with Ising models generalize very naturally to more complicated problems. However, it is sometimes the case that we seem to get “lucky” and find solutions which do not require

¹No offense to anyone whose problems have been left out.

expansions of the state space: for example, present a solution to a tricky minimax graph problem by a separation of three energy scales, although this trick does not seem to generalize as well as the others.

There has been debate about whether or not these algorithms would actually be useful for a quantum computer [5, 1], and this paper does not settle this debate. However, it may be useful in the distant future to have a quantum computer solve a large NP problem without the aid of a classical computation, and so therefore finding quantum Hamiltonians whose ground states correspond directly to the solutions of NP problems may be prove useful.

In a sense, the goal of this paper is trivial, as Ising spin glasses are known to be NP-hard [2], so it is natural to suspect connections with all other NP problems. This triviality is exaggerated by the fact that a simple NP-complete number partitioning problem is simply related to an Ising spin glass, so if there are polynomial time maps of all other NP-complete maps into this problem, we expect polynomial size Ising formulations. However, many of these direct constructions do not seem to be written down in a manner accessible for physicists or quantum algorithm designers, if they have been at all. Thus, the goal of this paper is to present these constructions in one place, in a self-contained manner, starting from trivial constructions and moving onwards to significantly more subtle constructions, and culminating in an Ising model for each of Karp's 21 NP-complete problems.

2. Partitioning Problems

The simplest problems to phrase as Ising models are partitioning problems: these maps are celebrated and well-known [8, 6]. For completeness, we review them here. Some of the techniques used in these models will prove useful in more nontrivial constructions.

2.1. Number Partitioning

Number partitioning asks the following: given a set S of numbers, which we will take to be natural numbers, is there a partition of this set of numbers into two disjoint subsets R and $S - R$, such that the sum of the elements in both sets is the same? This can be phrased trivially as an Ising model as follows. Let n_i ($i = 1, \dots, N = |S|$) describe the numbers in set S , and let

$$H = A \left(\sum_{i=1}^N n_i s_i \right)^2 \quad (2)$$

be an energy functional, where $s_i = \pm 1$ is an Ising spin variable. Here $A > 0$ is some positive constant. Typically, such constants are scaled to 1 in the literature, but for simplicity we will retain them, since in many formulations a separation of energy scales will prove useful and retaining each scale can make it easier to follow.

It is clear that if there is a solution to the Ising model with $H = 0$, then there is a configuration of spins where the sum of the n_i for the $+1$ spins is the same for the sum of the n_i for the -1 spins. Thus, if the ground state energy is $H = 0$, there is a solution to the number partitioning problem. Degeneracies in this ground state which do not correspond to $s_i^* \rightarrow -s_i^*$ correspond to distinct solutions to the partitioning problem. Furthermore, if the ground state has $H > 0$, we know that there are no solutions to the partitioning problem, but the ground state we do find is (one of) the best possible solutions, in the sense that it minimizes the mismatch.

2.2. Graph Partitioning

Graph partitioning is the classic example of a map between the physics of Ising spin glasses and NP problems. We will phrase it in a slightly different form than the original [6], which employed a hard constraint on the phase space. We will want none of our formulations to do this, as this may hamper their ability to be used in quantum computing applications.

Let us consider an undirected graph $G = (V, E)$ with an even number $N = |V|$ of vertices. We ask: what is a partition of the set V into two subsets of equal size $N/2$ such that the number of edges connecting the two subsets is minimized? This is known to be an NP-complete problem. As before, we will place an Ising spin $s_v = \pm 1$ on each vertex $v \in V$ on the graph, and we will let $+1$ and -1 denote the vertex being in either the $+$ set or the $-$ set. We solve this with an energy functional consisting of two components:

$$H = H_A + H_B \quad (3)$$

where

$$H_A = A \left(\sum_{n=1}^N s_i \right)^2 \quad (4)$$

is an energy which provides a penalty if the number of elements in the $+$ set is not equal to the number in the $-$ set, and

$$H_B = B \sum_{(uv) \in E} \frac{1 - s_u s_v}{2} \quad (5)$$

is a term which provides an energy penalty B for each time that an edge connects vertices from different subsets. If $B > 0$, then we wish to minimize the number of edges between the two subsets; if $B < 0$, we will choose to maximize this number. Should we choose $B < 0$, we must ensure that B is small enough so that it is never favorable to violate the constraint of H_A in order to minimize energy.

2.3. Cliques

A clique of size K in an undirected graph $G = (V, E)$ is a subset $W \subseteq V$ of the vertices, of size $|W| = K$, such that the subgraph (W, E_W) (where E_W is the edge set E restricted to edges between nodes in W) is a fully connected graph – i.e., all possible $K(K-1)/2$ edges in the graph are present.

We show how the NP-complete decision problem of whether or not a clique of size K exists can be written as an Ising-like model. We place a spin variable $s_v = \pm 1$ on each vertex $v \in V$ of the graph. In general, in this paper, for a spin variable s_α , we will define the binary bit variable

$$x_\alpha \equiv \frac{s_\alpha + 1}{2}. \quad (6)$$

It will typically be more convenient to phrase the energies in terms of this variable x_α , as it will be for this problem. Note that any energy functional which was quadratic in s_v will remain quadratic in x_v , and vice versa, so we are free to use either variable.

We will write the energy as

$$H = A \left(K - \sum_v x_v \right)^2 + A \left[\frac{K(K-1)}{2} - \sum_{(uv) \in E} x_u x_v \right] \quad (7)$$

where $A > 0$ is a positive constant. The ground state of this Hamiltonian is $H = 0$ if and only if a clique of size K exists. This is because the role of the first term is to enforce the constraint that exactly K vertices may be included in a trial subset W of the vertices, and the second term subsequently gives an

energy penalty for each possible edge between two vertices of W which is not included in E_W . There are no energy penalties from the latter term precisely when the trial set W forms a clique. Given a ground state solution, it is of course easy to read off from the x_v which K nodes form a clique.

A quantum algorithm can actually be made slightly more efficient so long as the initial state can be carefully prepared [3].

3. Hard Constraint Covering and Packing Problems

In this section, we will turn to what we call “hard constraint” problems related to covering and packing sets. These are essentially problems where constraints must be exactly satisfied. Many of the problems described below are often discussed in the literature, but again we review them here for completeness.

3.1. Binary Integer Linear Programming

Consider the problem, where x_1, \dots, x_N are binary variables, arranged into a vector \mathbf{x} . If we ask, what is the largest value of $\mathbf{c} \cdot \mathbf{x}$, for some vector \mathbf{c} , given a constraint

$$A\mathbf{x} = \mathbf{b} \quad (8)$$

with A some matrix and \mathbf{b} some vector with m components.

We solve this as followed. Let

$$H = H_A + H_B \quad (9)$$

where

$$H_A = A \sum_{j=1}^m \left[b_j - \sum_{i=1}^N A_{ji} x_i \right]^2 \quad (10)$$

and $A > 0$ is a constant. The ground states of $H_A = 0$ enforce (if such a ground state exists, of course!) the constraint that $A\mathbf{x} = \mathbf{b}$. Then we set

$$H_B = -B \sum_{i=1}^N c_i x_i. \quad (11)$$

with $B \ll A$ another positive constant. The energy scale separation is required so that it is never favorable to violate the H_A constraint.

3.2. Exact Cover

The exact cover problem goes as follows: consider a set $U = \{1, \dots, n\}$, and subsets $V_i \subseteq U$ such that

$$U = \bigcup_i V_i. \quad (12)$$

The question is: is there a subset of $\{V_i\}$, called R , such that the elements of R are disjoint sets, and the union of the elements of R is U ? This problem was solved in [4] but for simplicity, we repeat it here.

We write this problem as an Ising model as follows. Let us denote a spin variable s_i and binary variable x_i for each subset V_i . The Hamiltonian we use is $H = H_A + H_B$, where

$$H_A = A \sum_{\alpha=1}^n \left(1 - \sum_{i: \alpha \in V_i} x_i \right)^2. \quad (13)$$

$H_A = 0$ precisely when every element is included exactly one time, which implies that the unions are disjoint. The existence of a ground state of energy $H = 0$ corresponds to the existence of a solution to the exact cover problem.

It is also straightforward to extend this, and find the *smallest* exact cover. This is done by simply adding a second energy scale $B \ll A$:

$$H_B = B \sum_i x_i. \quad (14)$$

The ground state of this model will be nB , where n is the smallest number of subsets required.

3.3. Set Packing

Let us consider the same setup as the previous problem, but now ask a different question: what is the largest number of subsets V_i which are all disjoint? This is called the set packing problem.² To do this, we use $H = H_A + H_B$ with a separation of energy scales. We use

$$H_A = A \sum_{i,j: V_i \cap V_j \neq \emptyset} x_i x_j, \quad (15)$$

which is minimized only when all subsets are disjoint. Then, we use

$$H_B = -B \sum_i x_i \quad (16)$$

which simply counts the number of sets we included. Choosing $B < A$ ensures that it is never favorable to violate the constraint H_A (since there will always be a penalty of at least A per extra set included).

Note that an equivalent formulation of this problem is as follows: let us consider the sets to be encoded in an undirected graph $G = (V, E)$, where each set V_i corresponds to a vertex $i \in V$. An edge $ij \in E$ exists when $V_i \cap V_j$ is nonempty. It is straightforward to see that if we replace

$$H_A = A \sum_{ij \in E} x_i x_j \quad (17)$$

that the question of what is the maximal number of vertices which may be “colored” ($x_i = 1$) such that no two colored vertices are connected by an edge, is exactly equivalent to the set packing problem described above.

3.4. Vertex Cover

Given an undirected graph $G = (V, E)$, what is the smallest number of vertices that can be “colored” such that every edge is incident to a colored vertex? Let x_v be a binary variable on each vertex, which is 1 if it is colored, and 0 if it is not colored. Our Hamiltonian will be $H = H_A + H_B$. The constraint that every edge has at least colored vertex is encoded in H_A :

$$H_A = A \sum_{uv \in E} (1 - x_u)(1 - x_v). \quad (18)$$

Then, we want to minimize the number of colored vertices with H_B :

$$H_B = B \sum_v x_v \quad (19)$$

Choose $NB < A$, where $N = |V|$. Then the ground state of this Ising model corresponds to the configuration of colored vertices such that each edge connects to a colored vertex, and the fewest number of vertices are colored.

²Often times, this is also referred to as the maximally independent set (MIS) problem.

3.5. Satisfiability

Satisfiability is one of the most famous NP-complete problems. Every satisfiability problem can be written as a so-called 3SAT problem in conjunctive normal form (and this algorithm takes only polynomial steps/time) and so we will focus for simplicity on this case. In this case, we ask whether

$$\Psi = C_1 \wedge C_2 \cdots \wedge C_m \quad (20)$$

can take on the value of true – i.e., every C_i for $1 \leq i \leq m$ is true, where the form of each C_i is:

$$C_i = y_{i_1} \vee y_{i_2} \vee y_{i_3} \quad (21)$$

Here y_{i_1} , y_{i_2} and y_{i_3} are selected from another set of Boolean variables: $x_1, \dots, x_N, \bar{x}_1, \dots, \bar{x}_N$.

There is a well-known reduction of 3SAT to the set packing problem [4] which we reproduce here, for completeness. Consider solving the set packing problem on the following graph G with $3m$ nodes, which we construct as follows. For each clause C_i , we add 3 nodes to the graph, and connect each node to the other 3. After this step, if there is a y_1 and y_2 such that $y_1 = x_j$ and $y_2 = \bar{x}_j$, then we also add an edge between these two nodes. Solving the set packing problem on this G , and asking whether the solution has exactly m nodes, is equivalent to solving the 3SAT problem. This can be seen as follows: if a solution to the 3SAT problem exists, only one element of each clause needs to be true – if more are true, that is also acceptable, but we must have that one is true, so let us choose to color the vertex corresponding to the variable which is true. However, we may also not choose to have both x_1 be true and \bar{x}_1 be true, so we are required to connect all such points with an edge.

3.6. Minimax Graph Problem

The minimax graph problem proceeds as follows: let $G = (V, E)$ denote an undirected graph, and let $C \subseteq E$ be a proposed “coloring”. The constraints on C are as follows: for each edge in C , let us color the two vertices it connects. We will then demand that: no two edges in C share a vertex, and that there are no two uncolored vertices connected by an edge. This is minimal in that we cannot add any more edges to C (coloring any appropriate vertices) without violating the first constraint, and maximal in the sense that the trivial empty set solution is not allowed – we must include all edges between uncolored vertices.

Note that, from this point on in this paper, we have not found the Ising formulations of this paper in the literature.

We will use the spins on the graph to model whether or not an edge is colored. Let us use the binary variable x_e to denote whether or not an edge is colored. Our method consists of the following trick. Our energy will consist of a series of three energy functionals, each of which is chosen to be “much” smaller than the previous one:

$$H = H_A + H_B + H_C. \quad (22)$$

The first and largest term, H_A , will impose the constraint that no vertex has two colored edges. This can be done by setting

$$H_A = A \sum_v \sum_{\{e_1, e_2\} \subset \partial v} x_{e_1} x_{e_2}. \quad (23)$$

Here $A > 0$ is a positive energy, and ∂v corresponds to the subset of E of edges which connect to v . Thus the ground states consist of $H_A = 0$; if $H_A > 0$, it is because there is a vertex where two of its edges are colored.

If we choose A to be large enough, then we also can define, for states with $H_A = 0$, the variable

$$y_v \equiv \begin{cases} 1 & v \text{ has a colored edge} \\ 0 & v \text{ has no colored edges} \end{cases} = \sum_{e \in \partial v} x_e. \quad (24)$$

We stress that this definition is only valid for states with $H_A = 0$, since in these states each vertex has either 0 or 1 colored edges. We then define the energy H_B , such that solutions to the minimax coloring problem also have $H_B = 0$. Since we have already constrained the number of colored edges per vertex, we choose H_B to raise the energy of all solutions where there exists a possible edge which can be colored, yet still not violate the coloring condition, out of the ground state. To do this, we can sum over all edges in the graph, and check whether or not the edge connects two vertices, neither of which are colored:

$$H_B = B \sum_{e=(uv)} (1 - y_u)(1 - y_v). \quad (25)$$

Note that since, y_v can be negative, we must choose $B > 0$ to be small enough. This will ensure that a ground state will have energy $H_A + H_B = 0$, and correspond precisely to $H_A = H_B = 0$: i.e., states which do not violate the minimax constraints.

Now, given the states where $H_A = H_B = 0$, we now want the ground state to be the state where the fewest number of edges are colored. To do this, we simply let

$$H_C = C \sum_e x_e \quad (26)$$

count the number of colored edges. Here C is an energy scale chosen to be small enough such that it is never energetically favorable to violate the constraints imposed by either the H_A or H_B terms: for example, $CN < 2B$. The term with the smallest H_C has the smallest number of edges, and is clearly the solution to the minimax problem. Each ground state of this spin model is equivalent to a solution of the minimax problem.

4. Soft Constraint Packing and Covering Problems

We now turn to NP problems whose formulations as Ising models are substantially more subtle. This is, in every case, due to the fact that the constraints on the system are “soft” constraints which are only inequalities. For most problems of interest, these soft constraints can be re-written as “hard constraints” by an expansion of the number of spins.

4.1. Set Cover

Consider a set $U = \{1, \dots, n\}$, with sets $V_\alpha \subseteq U$ ($\alpha = 1, \dots, N$) such that

$$U = \bigcup_{\alpha=1}^N V_\alpha. \quad (27)$$

The set covering problem is to find the smallest possible number of V_α s, such that the union of them is equal to U .

Our solution consists of the following. Let us denote x_α to be a binary variable which is 1 if set α is included, and 0 if set α is not included. Let us then denote $x_{i,m}$ to be a binary variable which is 1 if the number of V_α s which include element i is $m \geq 1$, and 0 otherwise. Set $H = H_A + H_B$. Our first energy imposes the constraints that exactly one $x_{i,m}$ must be 1, since each element of U must be included a fixed number of times, and that the number of times that we claimed i was included is in fact equal to the number of V_α we have included, with i as an element:

$$H_A = A \sum_{i=1}^n \left(1 - \sum_{m=1}^N x_{i,m} \right)^2 + A \sum_{i=1}^n \left(\sum_{m=1}^N m x_{i,m} - \sum_{\alpha: i \in V_\alpha} x_\alpha \right)^2. \quad (28)$$

Finally, we minimize over the number of V_α s included:

$$H_B = B \sum_{\alpha} x_{\alpha}, \quad (29)$$

with $B \ll A$ chosen so that the A constraint is never violated.

4.2. Knapsack with Integer Weights

The knapsack problem is the following NP-complete problem: given a list of N objects, labeled by indices α , with the weight of each object given by w_α , and a knapsack which can only carry weight W , what is the heaviest collection of objects which we can add to the knapsack such that their combined weight is smaller than W ?

Let x_α be a binary variable denoting if we included α , and x_n for $1 \leq n \leq W$ denote a binary variable which is 1 if the final weight of the knapsack is n , and 0 otherwise. Our solution consists of letting $H = H_A + H_B$, with

$$H_A = A \left(1 - \sum_{n=1}^W x_n \right)^2 + A \left(\sum_{n=1}^W n x_n - \sum_{\alpha} w_{\alpha} x_{\alpha} \right)^2 \quad (30)$$

which enforces that the weight can only take on one value and that the weight of the objects in the knapsack equals the value we claimed it did, and finally

$$H_B = -B \sum_{n=1}^W n x_n, \quad (31)$$

with $B > 0$ and $B \ll A$, chosen such that it is never favorable to violate the constraints.

There is a trick which can be used to dramatically reduce the number of extra x_i spins which must be added. For simplicity, let us focus on the case $W = 2^M$, for M some positive integer, where the trick works most efficiently. In this case, we only need x_1, \dots, x_M instead of x_1, \dots, x_W . It is easy to check that

$$H = A \left(\sum_{n=1}^M 2^{n-1} x_n - \sum_{\alpha} w_{\alpha} x_{\alpha} \right)^2 - B \sum_{n=1}^M 2^{n-1} x_n \quad (32)$$

solves the exact same knapsack problem. This trick can be used on many of the problems below as well, but it makes the physical intuition a bit less clear, so we will not use it in our explicit formulations.

5. Coloring Problems

We now turn to coloring problems. Naively, coloring problems are often best phrased as Potts models, but these classical Potts models can be converted to classical Ising models with an expansion of the number of spins. This simple trick forms the basis for our solutions to this class of problems.

5.1. Graph Coloring

Given an undirected graph $G = (V, E)$, and a set of n colors, is it possible to color each vertex in the graph with a specific color, such that no edge connects two vertices of the same color? This is a generalization of the classic problem of how many colors are needed to color a map, such that every two countries which share a border have a different color, and is called the graph coloring problem.

Our solution consists of the following: we denote $x_{v,i}$ to be a binary variable which is 1 if vertex v is colored with color i , and 0 otherwise. The energy is

$$H = A \sum_v \left(1 - \sum_{i=1}^n x_{v,i} \right)^2 + A \sum_{(uv) \in W} \sum_{i=1}^n x_{u,i} x_{v,i}. \quad (33)$$

The first term enforces the constraint that each vertex has exactly one color, and provides an energy penalty each time this is violated, and the second term gives an energy penalty each time an edge connects two vertices of the same color. If there is a ground state of this model with $H = 0$, then there is a solution to the coloring problem on this graph with n colors. We can also read off the color of each node (in one such coloring scheme) by looking at which x s are 1. Note that the number of spins can be slightly reduced, since there is a permutation symmetry among colorings, by choosing a specific node in the graph to have the color 1, and one of its neighbors to have the color 2, for example.

5.2. Clique Cover

The clique cover problem, for an undirected graph $G = (V, E)$, is the following: given n colors, is there a coloring of the graph such that if we assign every vertex in the graph exactly one color, then the subgraph for any color is a clique?

Our solution is very similar to the graph coloring problem. Again, we employ the same binary variables as for graph coloring, and use Hamiltonian

$$H = A \sum_v \left(1 - \sum_{i=1}^n x_{v,i} \right)^2 + A \sum_{i=1}^n \left[\frac{1}{2} \left(-1 + \sum_v x_{v,i} \right) \sum_v x_{v,i} - \sum_{(uv) \in W} x_{u,i} x_{v,i} \right]. \quad (34)$$

The first term enforces the constraint that each vertex has exactly one color by giving an energy penalty each time this constraint is violated. In the second term, since the sum over v of $x_{v,i}$ counts the number of nodes with color i , the first sum counts highest possible number of edges that could exist with color i . The second term then checks if, in fact, this number of edges does in fact exist. Thus $H = 0$ if and only if the clique cover problem is solved by the given coloring. If a ground state exists with $H = 0$, there is a solution to the clique covering problem.

5.3. Job Sequencing with Integer Lengths

The job sequencing problem is as follows: given a list of N jobs for, say, a computer cluster, and job i has length L_i , how can each job be assigned to a computer in the cluster such that, if the length of cluster α , and its set of jobs is V_α , then

$$M_\alpha \equiv \sum_{i \in V_\alpha} L_i. \quad (35)$$

We assume that $L_i \in \mathbb{N}$.

To do this, we will begin by demanding that without loss of generality, $M_1 \geq M_\alpha$ for any α . Introduce the variables $x_{i,\alpha}$ which are 1 if job i is added to computer α , and 0 otherwise, and the variables $y_{n,\alpha}$ for $\alpha \neq 1$ and $n \geq 0$, which is 1 if the difference $M_1 - M_\alpha = n$. Then the Hamiltonian

$$H_A = A \sum_{i=1}^N \left(1 - \sum_{\alpha} x_{i,\alpha} \right)^2 + A \sum_{\alpha=1}^M \left(\sum_n n y_{n,\alpha} + \sum_i L_i (x_{i,\alpha} - x_{i,1}) \right)^2 \quad (36)$$

encodes that each job can be given to exactly one computer, and that no computer can have a longer total length than computer 1. To find the minimal maximal length M_1 , we just use

$$H_B = B \sum_i L_i x_{i,1}. \quad (37)$$

6. Hamiltonian Cycles

In this section, we describe the solution to the (undirected or directed) Hamiltonian cycles problem, and subsequently the traveling salesman problem, which for the Ising spin glass formulation, is a trivial extension.

6.1. Hamiltonian Cycles and Paths

Let $G = (V, E)$, and $N = |V|$. The graph can either be directed or undirected, and the solution will not change. The Hamiltonian path problem is as follows: starting at some node in the graph, can one travel along an edge, visiting other nodes in the graph, such that one can reach every single node in the graph without ever returning to the same node twice? The Hamiltonian cycles problem asks that, in addition, the traveler can return to the starting point from the last node he visits.

Without loss of generality, let us label the vertices $1, \dots, N$, and take the edge set (uv) to be directed – i.e., the order uv matters. It is trivial to extend to undirected graphs, by just considering a directed graph with (vu) added to the edge set whenever (uv) is added to the edge set. Our solution will use N^2 spins: binary bit variables $x_{v,i}$, where v represents the vertex and i represents its order in a prospective cycle. Our energy will have three components. The first two things we require is to enforce that every vertex can only appear once in a cycle, and that there must be a j^{th} node in the cycle for each j . Finally, for the nodes in our prospective ordering, if $x_{u,j}$ and $x_{v,j+1}$ are both 1, then $(uv) \in E$. Note that $N + 1$ should be read as 1, in the expressions below, if we are solving the cycles problem. These are encoded in the Hamiltonian:

$$H = A \sum_{v=1}^n \left(1 - \sum_{j=1}^N x_{v,j} \right)^2 + A \sum_{j=1}^n \left(1 - \sum_{v=1}^N x_{v,j} \right)^2 + A \sum_{(uv) \notin E} \sum_{j=1}^N x_{u,j} x_{v,j+1}. \quad (38)$$

$A > 0$ is a constant. It is clear that a ground state of this system has $H = 0$ only if we have an ordering of vertices where each vertex is only included once, and adjacent vertices in the cycle have edges on the graph – i.e., we have a Hamiltonian cycle.

To solve Hamiltonian path, instead of Hamiltonian cycle, all we have to do is restrict the sum over j above from 1 to $N - 1$. Thus it does not care about whether or not the first and last nodes are also connected.

We note that it is straightforward to imagine slightly reducing the size of the state space for the Hamiltonian cycles problem as follows: it is clear that node 1 must always be included in a Hamiltonian cycle, and without loss of generality we can set $x_{1,1} = 1$: this just means that the overall ordering of the cycle is chosen so that node 1 comes first. This reduces the number of spins to $(N - 1)^2$.

6.2. Traveling Salesman

The traveling salesman problem for a graph $G = (V, E)$, where each edge uv in the graph has a weight W_{uv} associated to it, is to find the Hamiltonian cycle such that the sum of the weights of each edge in the cycle is minimized. Typically, the traveling salesman problem assumes a complete graph, but we have the technology developed to solve it on a more arbitrary graph.

To solve this problem, we use $H = H_A + H_B$, with H_A the Hamiltonian given for the directed (or undirected, if the graph is undirected for traveling salesman) Hamiltonian cycles problem. We then simply add

$$H_B = B \sum_{(uv) \in E} W_{uv} \sum_{j=1}^N x_{u,j} x_{v,j+1}. \quad (39)$$

with B small enough that it is never favorable to violate the constraints of H_A . If the traveling salesman does not have to return to his starting position, we can restrict the sum over j from 1 to $N - 1$, as before.

7. Tree Problems

The most subtle NP problems to solve with Ising models are problems which require finding connected tree subgraphs of larger graphs. The key point which makes the solution difficult is the requirement of connectivity, and the solution relies on similar tricks, therefore, to the Hamiltonian cycles problem discussed earlier.

7.1. Minimal Spanning Tree with a Maximal Degree Constraint

The minimal spanning tree is the following: given an undirected graph $G = (V, E)$, where each edge $(uv) \in E$ comes with a cost c_{uv} , what is the tree $T \subseteq G$, which contains all vertices, such that the cost of T , defined as

$$c(T) \equiv \sum_{(uv) \in E_T} c_{uv}, \quad (40)$$

is minimized? Without loss of generality, we take $c_{uv} > 0$ in this subsection (a large positive constant can always be added to each c_{uv} ensure that the smallest value of c_{uv} is strictly positive, since the number of edges of a tree is always $N - 1$, if the number of vertices in N). We will also add a degree constraint, that each degree in T be $\leq \Delta$, which makes the problem NP-complete.

To solve this problem, we place a binary variable y_e on each edge to determine whether or not that edge is included in T :

$$y_e \equiv \begin{cases} 1 & e \in E_T \\ 0 & \text{otherwise} \end{cases}. \quad (41)$$

We also place a large number of binary variables $x_{v,i}$ on each vertex, and $x_{uv,i} \neq x_{vu,i}$ on edge (uv) : the number $i = 0, 1, \dots, N/2$ will be used to keep track of the depth a node in the tree, and if $x_{uv} = 1$, it means that u is closer to the root than v , and if $x_{vu} = 1$ it means that v is closer to the root, and we use another variable $z_{v,i}$ ($i = 1, \dots, \Delta$) to count the number of degrees of each node. We now use energy $H = H_A + H_B$, where the terms in H_A are used to impose the constraints that: there is exactly one root to the tree, each vertex has a depth, each bond has a depth, and its two vertices must be at different heights, the tree is connected (i.e., exactly one edge to a non-root vertex comes from a vertex at lower depth), each node can have at most Δ edges, and each edge at depth i points between a node at depth

$i - 1$ and i , respectively:

$$\begin{aligned}
H_A = & A \left(1 - \sum_v x_{v,0} \right)^2 + A \sum_v \left(1 - \sum_i x_{v,i} \right)^2 + A \sum_{uv \in E} \left(y_{uv} - \sum_i (x_{uv,i} + x_{vu,i}) \right)^2 \\
& + A \sum_v \sum_{i=1}^{N/2} \left(x_{v,i} - \sum_{(uv) \in E} x_{uv,i} \right)^2 + A \sum_v \sum_{i=1}^{N/2} \left(\sum_{j=1}^{\Delta} j z_{v,j} - \sum_{uv, vu \in E} \sum_i x_{uv,i} \right)^2 \\
& + A \sum_{uv, vu \in E} \sum_{i=1}^{N/2} x_{uv,i} (2 - x_{u,i-1} - x_{v,i})
\end{aligned} \tag{42}$$

The ground states with $H_A = 0$ are trees which include every vertex. Note that when we sum over $uv \in E$, we are not counting vu as distinct, but when we notate $uv, vu \in E$, we *do* treat $uv \neq vu$. We then add

$$H_B = B \sum_{uv, vu \in E} \sum_{i=1}^{N/2} c_{uv} x_{uv,i}. \tag{43}$$

The minimum of E will find the minimal spanning tree, subject to the degree constraint.

7.2. Steiner Trees

The Steiner tree problem is somewhat similar to the problem above: given our costs c_{uv} , we want to find a minimal spanning tree for a subset $U \subset V$ of the vertices, with no degree constraints. To do this, we use the same Hamiltonian as for the minimal spanning tree, except we add binary variables y_v for $v \notin U$ which determine whether or not a node U is included, and use the Hamiltonian $H = H_A + H_B$, where H_A enforces constraints similarly to in the previous case:

$$\begin{aligned}
H_A = & A \left(1 - \sum_v x_{v,0} \right)^2 + A \sum_v \left(y_v - \sum_i x_{v,i} \right)^2 + A \sum_{uv \in E} \left(y_{uv} - \sum_i (x_{uv,i} + x_{vu,i}) \right)^2 \\
& + A \sum_v \sum_{i=1}^{N/2} \left(x_{v,i} - \sum_{(uv) \in E} x_{uv,i} \right)^2 + A \sum_{uv, vu \in E} \sum_{i=1}^{N/2} x_{uv,i} (2 - x_{u,i-1} - x_{v,i})
\end{aligned} \tag{44}$$

We then use H_B from the previous model to determine the minimum weight tree.

7.3. Directed Feedback Vertex Set

A feedback vertex set for a directed graph $G = (V, E)$ is a subset $F \subset V$ such that the subgraph $(V - F, \partial(V - F))$ is acyclic. We will refer to F as the feedback set. Solving a decision problem for whether or not a feedback set exists for $|F| \leq k$ is NP-complete. We solve the optimization problem of finding the smallest size of the feedback set first for a directed graph – the extension to an undirected graph will be a bit more involved.

Before solving this problem, it will help to prove two lemmas. The first lemma is quite simple: there exists a node in a directed acyclic graph which is not the end point of any edges. For suppose that for each vertex, there was an edge that ends on that vertex. Then pick an arbitrary vertex, pick any edge ending on that vertex, and follow that edge in reverse to the starting vertex. Repeat this process more than N times, and a simple counting argument implies that we must have visited the same node more than once, at least once. Thus, we have traversed a cycle in reverse, which contradicts our assumption.

The second lemma is as follows: a directed graph $G = (V, E)$ is acyclic if and only if there is a height function $h : V \rightarrow \mathbb{N}$ such that if $uv \in E$, $h(u) < h(v)$: i.e., every edge points from a node at lower height to one at higher height. That height function existence implies acyclic is easiest to prove using the contrapositive: suppose that a graph is cyclic. Then on a cycle of edges, we have

$$0 < \sum [h(u_{i+1}) - h(u_i)] = h(u_1) - h(u_n) + h(u_n) - h(u_{n-1}) + \cdots - h(u_1) = 0 \quad (45)$$

is a contradiction. To prove that an acyclic graph has a height function, we construct one recursively. Using our first lemma, we know that there exists a vertex u with only outgoing edges, so let us call $h(u) = 1$. For any other vertex, we will call the height of that vertex $h(v) = 1 + h'(v)$, where $h'(v)$ is found by repeating this process on the graph with node u removed (which must also be acyclic). It is clear this process will terminate and assign exactly one node height i for each integer $1 \leq i \leq |V|$.

We can now exploit this lemma to write down an Ising spin formulation of this problem. We place a binary variable y_v on each vertex, which is 0 if v is part of the feedback set, and 1 otherwise. We then place a binary variable $x_{v,i}$ on each vertex, which is 1 if vertex v is at height i . So far the heights i are arbitrary, and the requirement that a height function be valid will be imposed by the energy. The energy functional we use is $H = H_A + H_B$ where

$$H_A = A \sum_v \left(y_v - \sum_i x_{v,i} \right)^2 + A \sum_{uv \in E} \sum_{i \geq j} x_{u,i} x_{v,j}. \quad (46)$$

The first term ensures that if a vertex is not part of the feedback set, it has a well-defined height; the second term ensures that an edge only connects a node with lower height to a node at higher height. We then find the smallest possible feedback set by adding

$$H_B = B \sum_v (1 - y_v). \quad (47)$$

7.4. Undirected Feedback Vertex Set

The extension to undirected graphs requires a bit more care. In this case, we have to be careful because there is no a priori distinction on whether the height of one end of an edge is smaller or larger than the other – this makes the problem much more involved, at first sight. Furthermore, it is not true that a directed acyclic graph is acyclic if the orientation of edges is ignored. However, for an undirected graph, we also know that a feedback vertex set must reduce the graph to trees, although there is no longer a requirement that these trees are connected (this is called a forest). With this in mind, we find that the problem is actually extremely similar to minimal spanning tree, but without degree constraints or connectivity constraints. The new subtlety, however, is that we cannot remove edges.

To solve this problem, we do the following: introduce a binary variable $x_{v,i}$, which is 1 if v is a vertex in any tree at depth i , and 0 otherwise. However, to account for the fact that we may remove vertices, we will allow for $x_{v,-1} = 1$ if v is part of the feedback vertex set, and 0 otherwise. We do a similar thing for edges: we consider $x_{uv,i} \neq x_{vu,i}$ to be defined as before when $i > 0$. We also define the variables $x_{uv,-1} \neq x_{vu,-1}$, which we take to be 1 when the ending node of the “directed” edge is in the feedback vertex set. Now, we can write down a very similar energy to the minimal spanning tree:

$$\begin{aligned} H_A = & A \sum_v \left(1 - \sum_i x_{v,i} \right)^2 + A \sum_{uv \in E} \left(1 - \sum_i (x_{uv,i} + x_{vu,i}) \right)^2 + A \sum_{uv \in E} (x_{uv,-1} - x_{v,-1})^2 \\ & + A \sum_v \sum_{i > 0} \left(x_{v,i} - \sum_{uv \in E} x_{uv,i} \right)^2 + A \sum_{uv, vu \in E} \sum_{i > 0} x_{uv,i} (2 - x_{u,i-1} - x_{v,i}) \end{aligned} \quad (48)$$

The changes are as follows: we no longer constrain only 1 node to be the root, or constrain the degree of a vertex – however, we have to add a new term to ensure that edges are only ignored in the tree constraint if they point to a node in the feedback set. We then add

$$H_B = B \sum_v x_{v,-1} \quad (49)$$

with $B \ll A$ chosen so that the A constraints are never violated. This counts the number of nodes in the feedback set, so thus H is minimized when H_B is smallest – i.e., we have to remove the fewest number of nodes.

7.5. Feedback Edge Set

For a directed graph, the feedback edge set problem is to find the smallest set of edges $F \subset E$ such that $(V, E - F)$ is a directed acyclic graph. It is known to be NP-hard.³ Our solution will be somewhat similar to the directed feedback vertex set. We place a binary variable y_{uv} on each edge, which is 1 if $uv \notin F$, and define $x_{uv,i}$ to be 1 if uv if both $y_{uv} = 1$ and the height of node u is i . We also add a binary variable $x_{v,i}$, as for the feedback vertex set. Our constraint energy must then enforce that: each vertex and included edge has a well-defined height, and that each edge points from a lower height to a higher height:

$$H_A = A \sum_v \left(1 - \sum_i x_{v,i}\right)^2 + A \sum_{uv \in E} \left(y_{uv} - \sum_i x_{uv,i}\right)^2 + A \sum_{uv} \sum_i x_{uv,i} \left(2 - x_{u,i} - \sum_{j>i} x_{v,j}\right). \quad (50)$$

We then use

$$H_B = B \sum_{uv \in E} (1 - y_{uv}) \quad (51)$$

to count the number of edges in F – it is minimized when this number is smallest.

8. Graph Isomorphisms

The question of whether two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are isomorphic is believed to be hard, but its classification into a complexity class is still a mystery. Since it is a hard problem, let us nonetheless describe an Ising formulation for it. An isomorphism is only possible if $|V_1| = |V_2| \equiv N$, so we will restrict ourselves to this case, and without loss of generality, we label the vertices of G_1 with $1, \dots, N$.

We write this as an Ising model as follows. Let us describe a proposed isomorphism through binary variables $x_{v,i}$ which is 1 if vertex v in G_2 gets mapped to vertex i in G_1 . The energy

$$H_A = A \sum_v \left(1 - \sum_i x_{v,i}\right)^2 + A \sum_i \left(1 - \sum_v x_{v,i}\right)^2 \quad (52)$$

ensures that this map is bijective. We then use an energy

$$H_B = B \sum_{ij \in E_1} \left(1 - \sum_{uv \in E_2} x_{u,i} x_{v,j}\right) + B \sum_{ij \notin E_1} \sum_{uv \in E_2} x_{u,i} x_{v,j} \quad (53)$$

to penalize a bad mapping: i.e. an edge that is not in G_1 is in G_2 , or an edge that is in G_1 is not in G_2 . If the ground state of this Hamiltonian corresponds to $H = 0$, there is an isomorphism.

³It is in P if the graph is undirected however.

9. Conclusion

In this paper, we have presented classical Ising formulations for a wide variety of famous NP problems. Although most of the new constructions are quite cumbersome and unlikely to be useful for a long time, if ever, the tricks introduced may be able to be refined and made efficient enough to become useful. Much of this paper can be thought of as an amusing exercise to simply enumerate many subtle constructions, although it may also turn out that some of the new constructions are quite useful for quantum algorithm designers as well. We stress that techniques involving separations of multiple scales are quite likely the most efficient for solving nontrivial problems, and this may be a useful direction to explore further.

We note that many of the Ising formulations of tree problems in this paper required substantial expansions of the state space and perhaps there are ways to avoid this with more clever Hamiltonians. I am unsure if it is an open question to determine the smallest possible scaling of state space size which can encode arbitrary instances of many NP problems, and this may be a worthwhile future direction.

The fact that there is an Ising model representation for a problem by no means ensure it is NP-complete or NP-hard. As a reminder that sometimes the Ising approach may be quite inconvenient, let us consider the following example. Consider the simple problem of finding the largest integer in a list n_1, \dots, n_N . Introducing binary variables x_i for $i = 1, \dots, N$, the Ising model

$$H = A \left(1 - \sum_i x_i \right)^2 - B \sum_i n_i x_i \quad (54)$$

for $A > B \max(n_i)$ solves this problem. In fact, this problem looks somewhat like an instance of the random field Ising model on a complete graph, and yet this has a very simple $O(N)$ classical algorithm. Thus, the results of this paper should be taken with a grain of salt – it is quite possible there are far more efficient algorithms, even for a quantum computer.

Acknowledgements

A.L. is supported by the Purcell Fellowship at Harvard.

He would like to thank Robert Lucas for suggesting this problem, and Vicky Choi, Jacob Sanders and John Tran for helpful comments.

References

- [1] V. Bapst, L. Foini, F. Krzakala, G. Semerjian, and F. Zamponi. “The quantum adiabatic algorithm applied to random optimization problems: the quantum spin glass perspective”, *Physics Reports* **523** (2013) [1210.0811].
- [2] F. Barahona. “On the computational complexity of Ising spin glass models”, *Journal of Physics* **A15** (1982).
- [3] A.M. Childs, E. Farhi, J. Goldstone, and S. Gutmann. “Finding cliques by quantum adiabatic evolution”, *Quantum Information and Computation* **2** (2002) [quant-ph/0012104].
- [4] V. Choi. “Adiabatic quantum algorithms for the NP-complete maximum-weight independent set, exact cover and 3SAT problems”, [1004.2226].

- [5] N.G. Dickson and M.H.S. Amin. “Does adiabatic quantum optimization fail for NP-complete problems?”, *Physical Review Letters* **106** (2011) [1010.0669].
- [6] Y. Fu and P.W. Anderson. “Application of statistical mechanics to NP-complete problems in combinatorial optimisation”, *Journal of Physics* **A19** (1986).
- [7] M.R. Garey and D.S. Johnson. Computers and Intractability: a Guide to the Theory of NP-Completeness (1979).
- [8] M. Mézard and A. Montanari. Information, Physics and Computation (2009).